Future Proof Parallelism for Electron-atom Scattering Codes on the Cray XT4

Andrew Sunderland, Mike Ashworth,
Martin Plummer, Cliff Noble

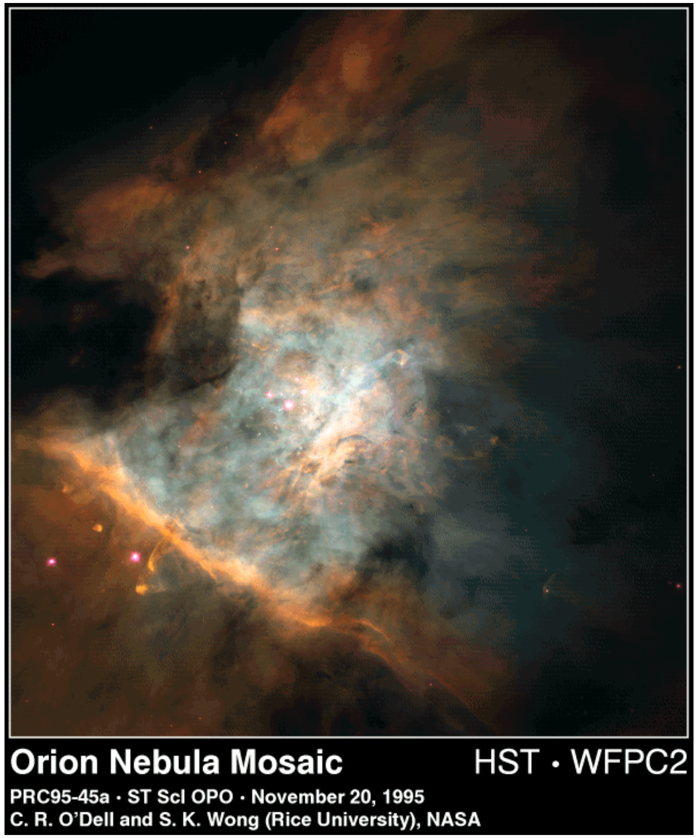
Advanced Research Computing and Atomic and
Molecular Physics Groups

STFC Daresbury Laboratory

Summary

- Background to R-matrix approach
  - Based on Baluja-Burke Morgan method
  - Results of interest
- Background to PFARM code
  - Design Features
  - Performance characterization
- EPSRC DCSE project for code optimization on Hector
  - Initial performance analysis on the XT4
  - Optimizations performed to date
  - Ongoing / future plans to expand the code

Electron-Atom Collisions

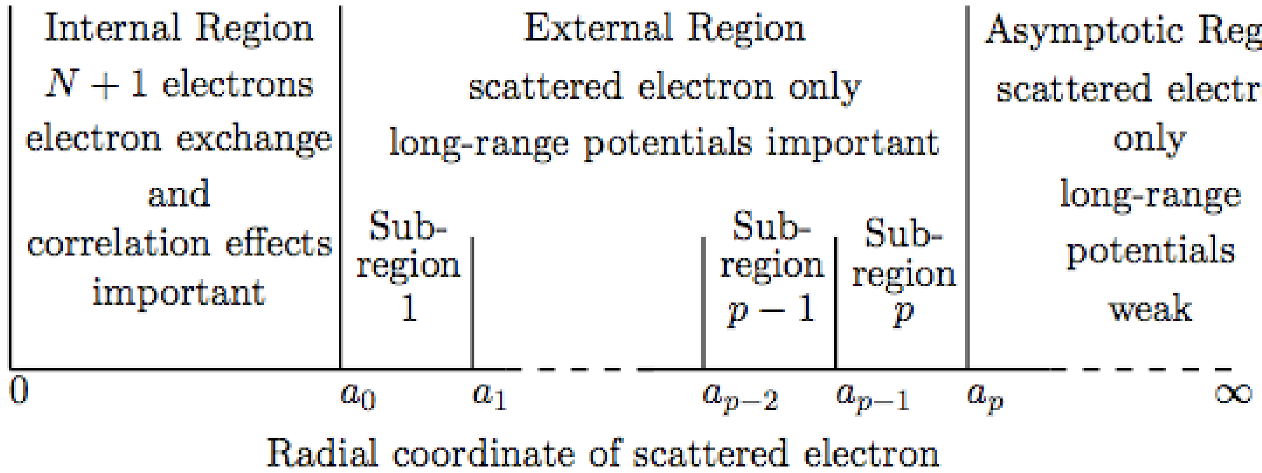


- Detailed electron-atom collision data is essential for understanding the behaviour of plasmas such as
  - Identifying forbidden lines such as those corresponding to the excitation of Ni^+ seen in observations of the Orion nebula (NGC 1976).
  - Plasma diagnostics of impurities in plasma fusion tokamaks.
  - Tin ions in next-generation nanolithography tools.
- R-matrix theory provides efficient computational methods for investigating electron-atom and electron-molecule collisions.
- Calculation involves integration of very large sets of coupled second-order linear differential equations. This presents huge computational challenges.

R-matrix Theory

- Basis of computer programs that describe a wide range of atomic, molecular, optical and surface processes
  - Past success: the International Opacity Project
- Successful in treating a wide range of collision phenomena
  - Scattering of electrons, positrons or photons with atomic and molecular targets.
- Numerically very stable
- Success has led to demand for more challenging cases
  - Interactions at intermediate energies
  - Open 3d-shell targets (iron-peak elements Fe, Co, Ni).
  - Open 4d-shell targets: eg tin ions, molybdenum (for environmentally friendly lighting)

Partition of Configuration Space



The parallelization of the code maps closely to this partitioning

Baluja-Burke-Morgan Method in the External Region Calculation

Used to solve the non-relativistic Schrodinger equation:

$$\mathbf{H}_{\mathbf{N+1}}\Psi = E\,\Psi$$

- R-matrices (inverse log-derivative matrices) at successively larger radial distances are obtained using Green's functions defined within finite radial sectors
- Green's functions are obtained using a shifted-Legendre basis.
- Diagonalize representative of the Green's function $(H + L - EI)^{-1}$ within a basis.

R-Matrix Propagation Program For Solving Coupled Second-order Differential Equations, K.L. Baluja, P.G.Burke and L.A.Morgan, Computer Physics Communications 27 (1982) 299-307

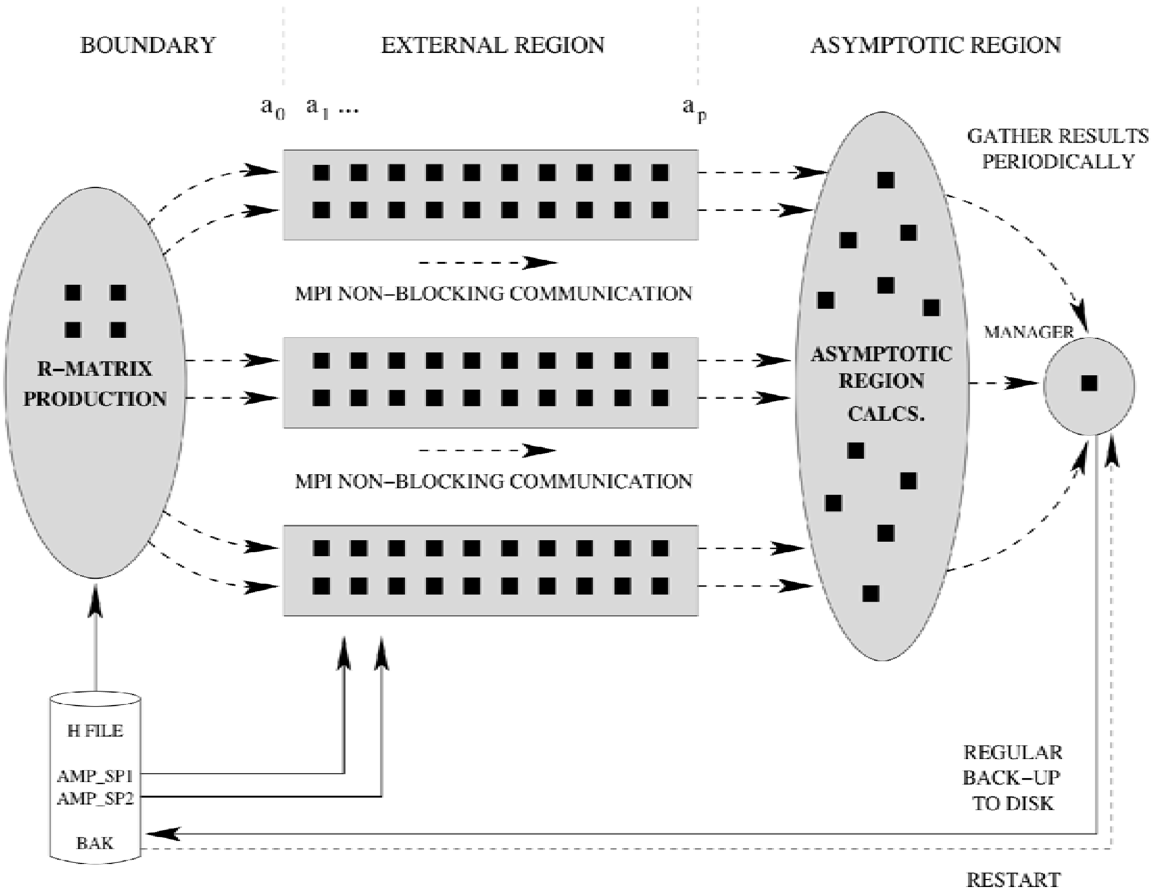
Science & Technology Facilities Council

Baluja-Burke-Morgan (BBM)-based Implementation

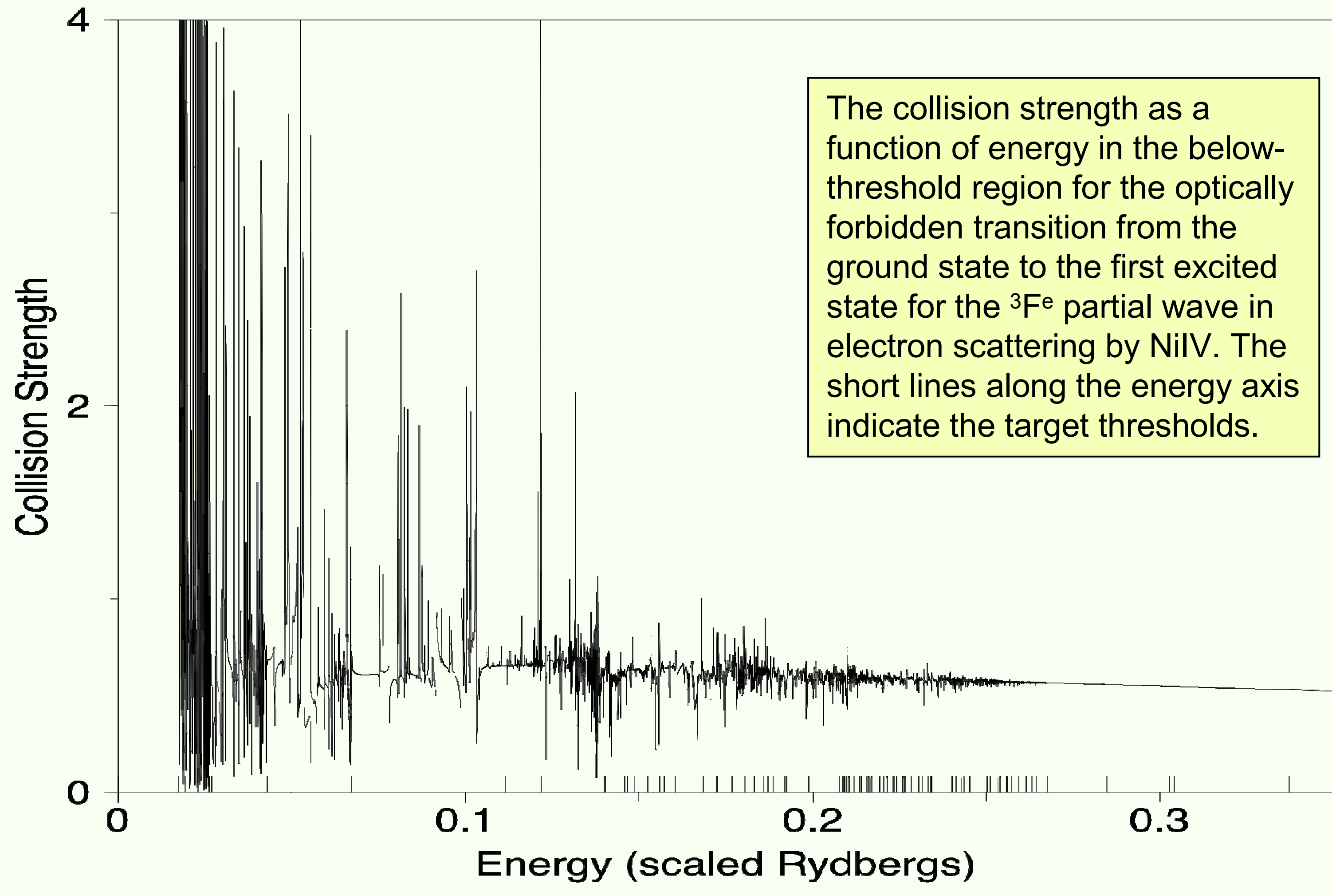
2 Stage Parallelization of BBM approach in the external region:

- EXDIG Program:
  - Diagonalize Sector Hamiltonian matrices using ScaLAPACK (Blacs-based Data decomposition).
- EXAS Program:
  - For each scattering energy propagate using 3 functional groups:
  - Generate initial R-Matrix (Data decomposition).
  - Propagate R-Matrix across each sector in pipeline (Control decomposition).
  - Calculate thermally averaged collision strengths (Task Farmed).

EXAS Parallelization



Results: Collision Strengths

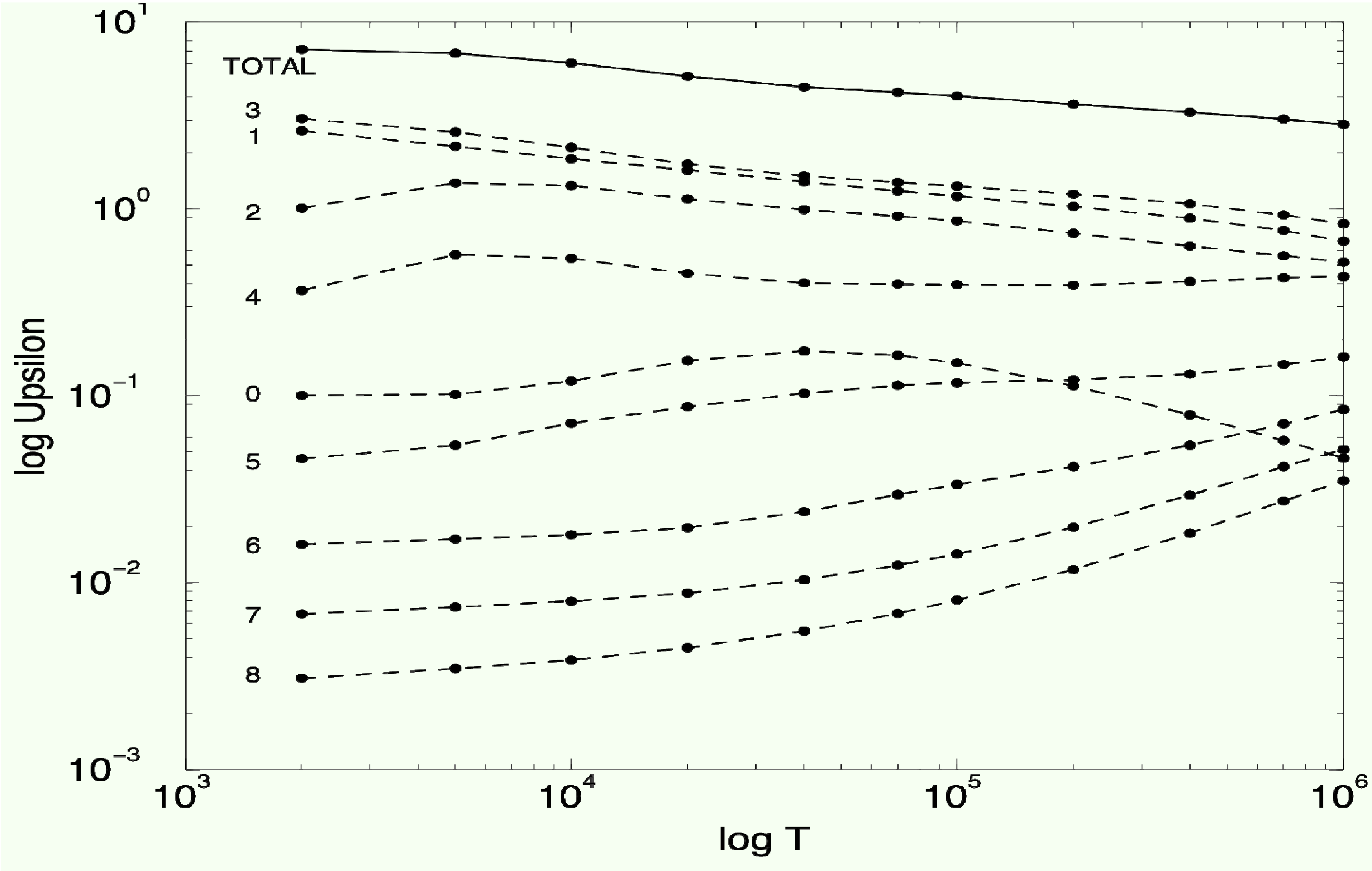


The collision strength as a function of energy in the below-threshold region for the optically forbidden transition from the ground state to the first excited state for the $^3F^e$ partial wave in electron scattering by NiIV. The short lines along the energy axis indicate the target thresholds.

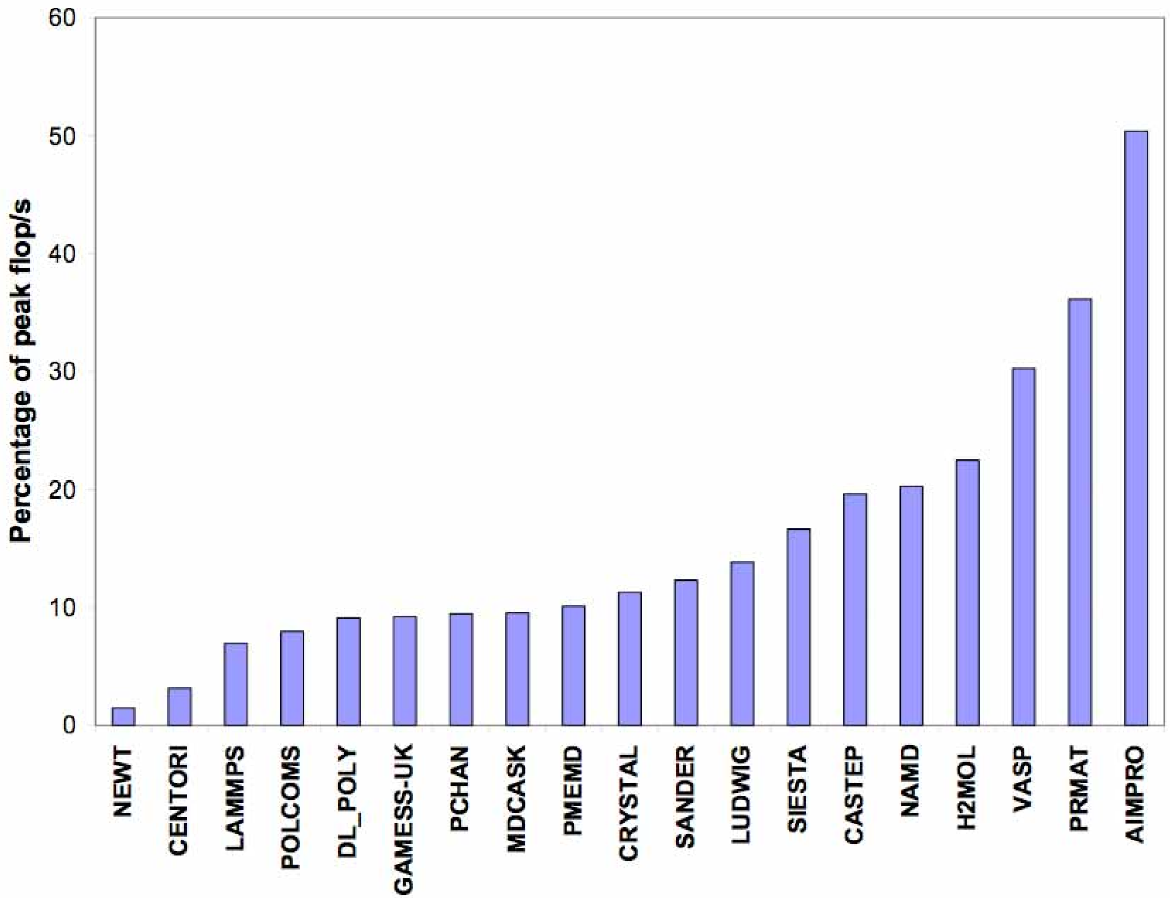
Thermally Averaged Collision Strengths NiIV

$$\Upsilon(T) = \int_0^{E_{max}} \Omega(E) e^{-\frac{E}{KT}} d\left(\frac{E}{KT}\right)$$

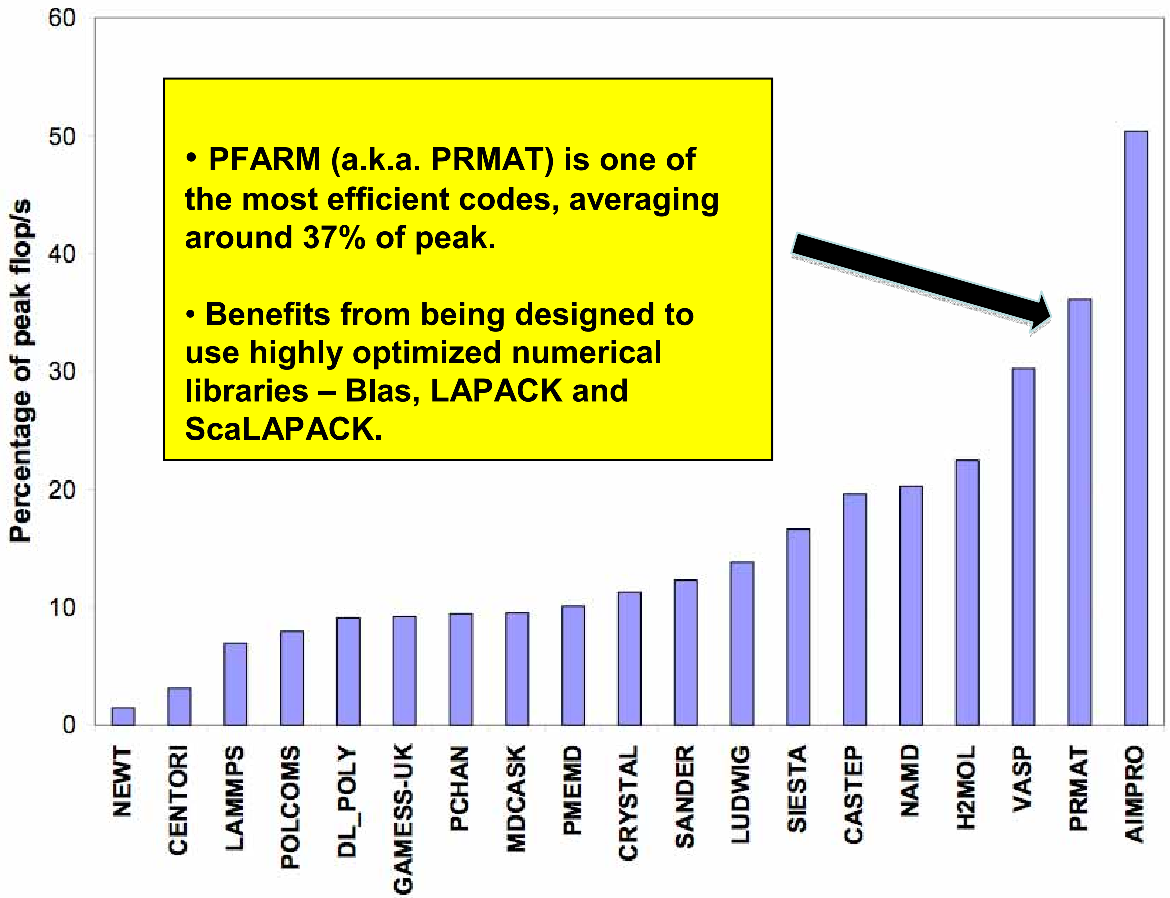
Dashed curves show the contribution to the effective collision strength for the optically forbidden transition from the ground state $^4F^e$ to the first excited state $^4P^e$ from individual partial waves from L=0 to L=8. The full curve shows the total.



Single Node Efficiency Study on HPCx (IBM PWR5)



Single Node Efficiency Study on HPCx (IBM PWR5)



Features of 2-stage PFARM Code

Scalable performance for a wide range of problem sizes:

- Replication of propagation pipelines (facilitated by MPI Groups and MPI Communicators)
- Fully flexible configuration to achieve optimal load balancing.
- Asynchronous characteristics permit effective overlapping of communication and computation (MPI Asynchronous, NonBlocking Send/Receive).
- Code is developed using standard Fortran95, MPI and numerical library routines from BLAS, LAPACK, PBLAS, ScaLAPACK
  - Produces portable and optimized code

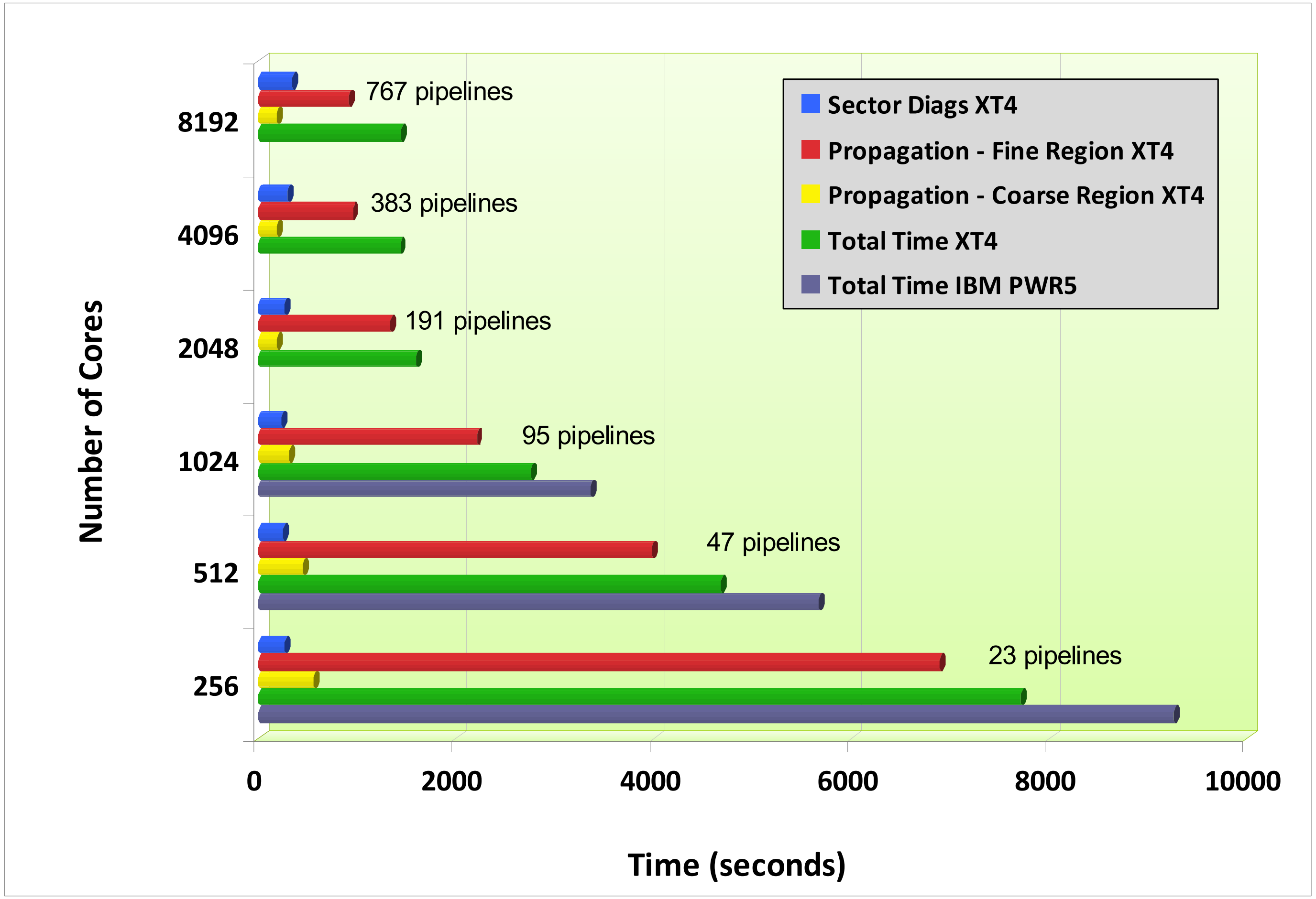
Physics-based optimizations: Scattering Energy Mesh Partition & Spin-splitting

- Vast majority of scattering energy points reside in the *fine* section of the energy mesh in the resonance region below the final threshold.
  - Highest energy in the *coarse* section of the energy mesh is usually around 4 *x* highest scattering energy in the fine section.
  - **Separate the fine and coarse mesh calculations** and maximize sector length, thereby reducing pipeline processors for each run.

- **Decouple channels associated with targets with different spin** *'spin-splitting'*.
  - Results in two smaller streams of data through the external region calculation thereby reducing computational and memory requirements

Initial Performance on Hector XT4

FeIIIJ030, 10667 scattering energies



EPSRC funded DSCE Project

- *"Future-proof parallelism for the electron-atom scattering codes PRMAT on HECToR for atomic physics and applications in industrial modelling and astrophysics"*
- 18 months funding to optimize and develop R-matrix codes on Hector Cray XT4
  - PFARM originally designed for Cray T3E with single core CPUs and limited memory. Certain assumptions made in the design are now out-dated and the code needs developing for modern systems.
  - Alternative approaches to BBM are also being developed by Cliff Noble
    - Airy LD propagator
      - Results in much reduced size of propagation matrices
      - *Alexander and Manopolous (J. Chem. Phys. 86 (1987), 2044-2050*

Optimization (i) – Sector Hamiltonian diagonalization

The 1st stage of the calculation EXDIG involves diagonalizing multiple sector Hamiltonian matrices:

– Dimension of sector Hamiltonian matrix can vary from 5000-100000+ depending upon the problem.

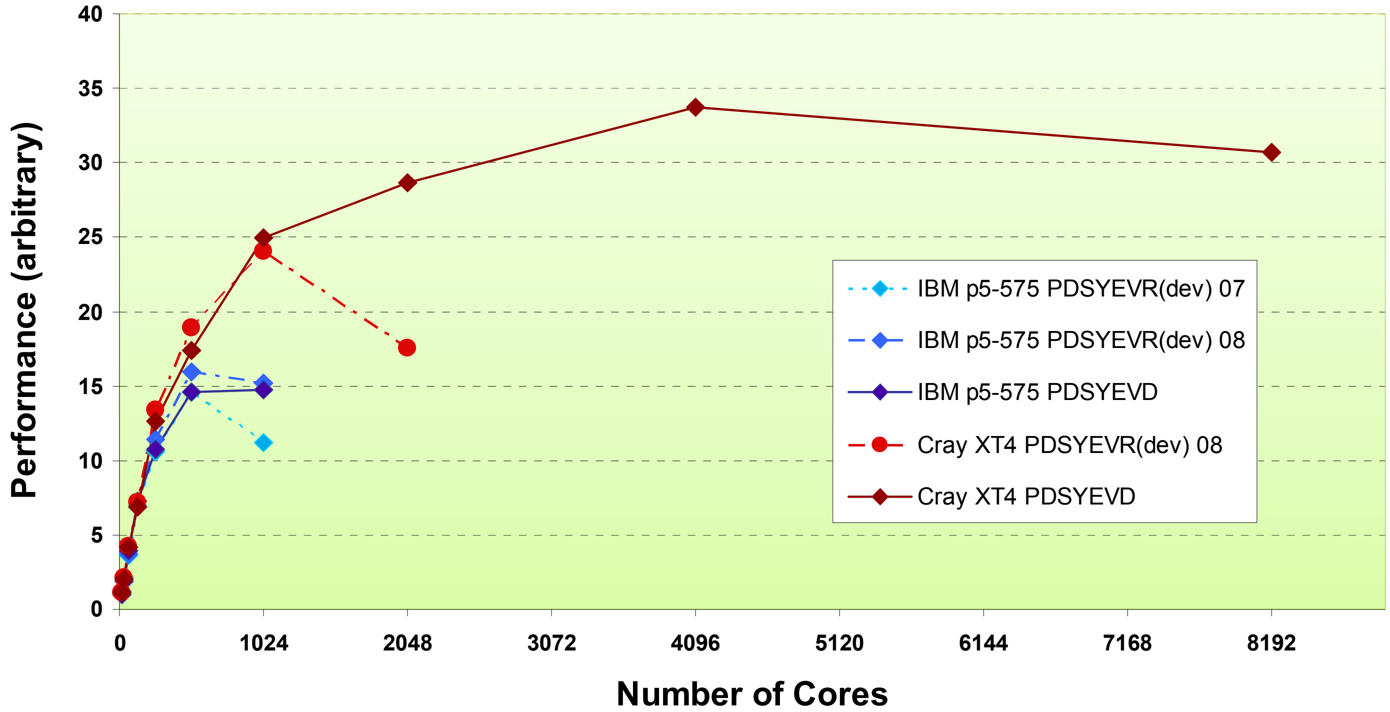
– Code currently steps through the sectors, undertaking a parallel diagonalization on each Hamiltonian using ScaLAPACK routines using **all the processors**.

– Optimizations:

1. Performance analysis of available diagonalization routines to determine best method (e.g. Divide-and-Conquer vs Multiple Relatively Robust Representation)
2. Divide the processors into sub-groups (split Blacs grids)
   - Sector matrices are distributed to each **sub-group where a parallel diagonalization** takes place independently of other sub-groups.

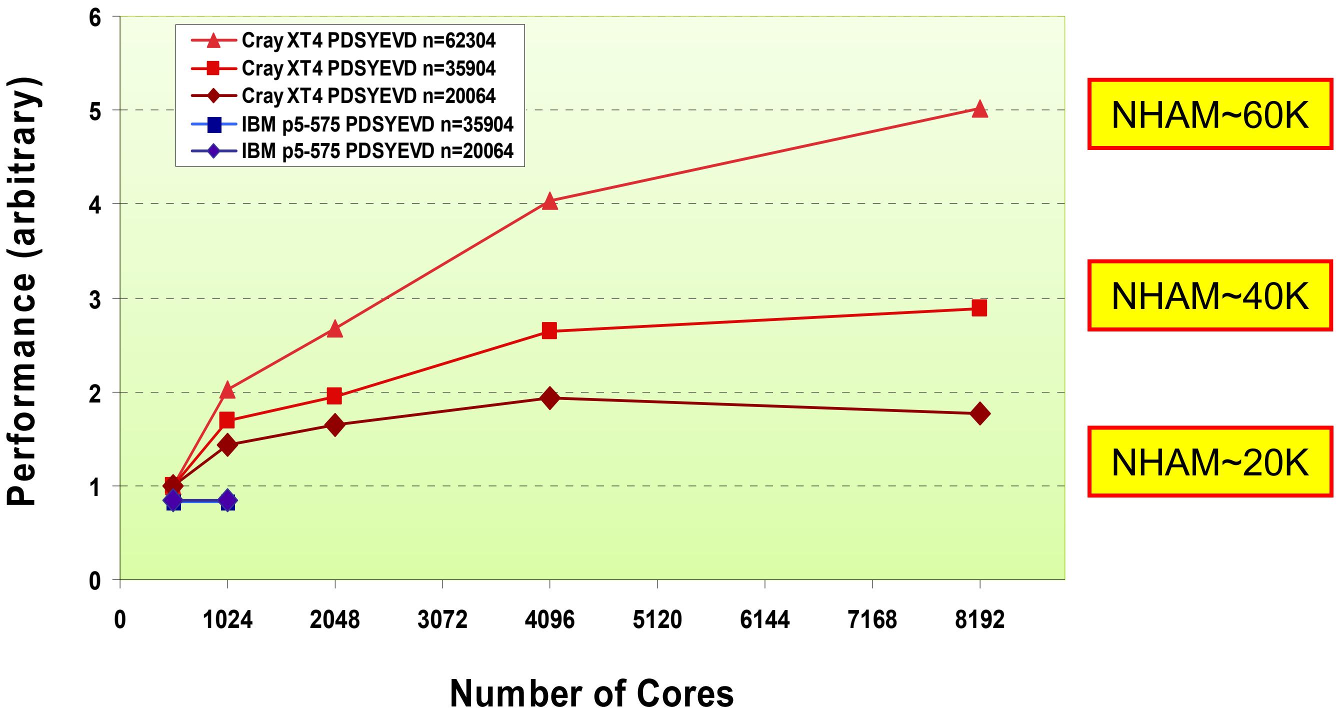
Parallel Diagonalization Performance Analysis

FeIII case NHAM=20064



Parallel Diagonalization Performance Analysis for a range of problem sizes

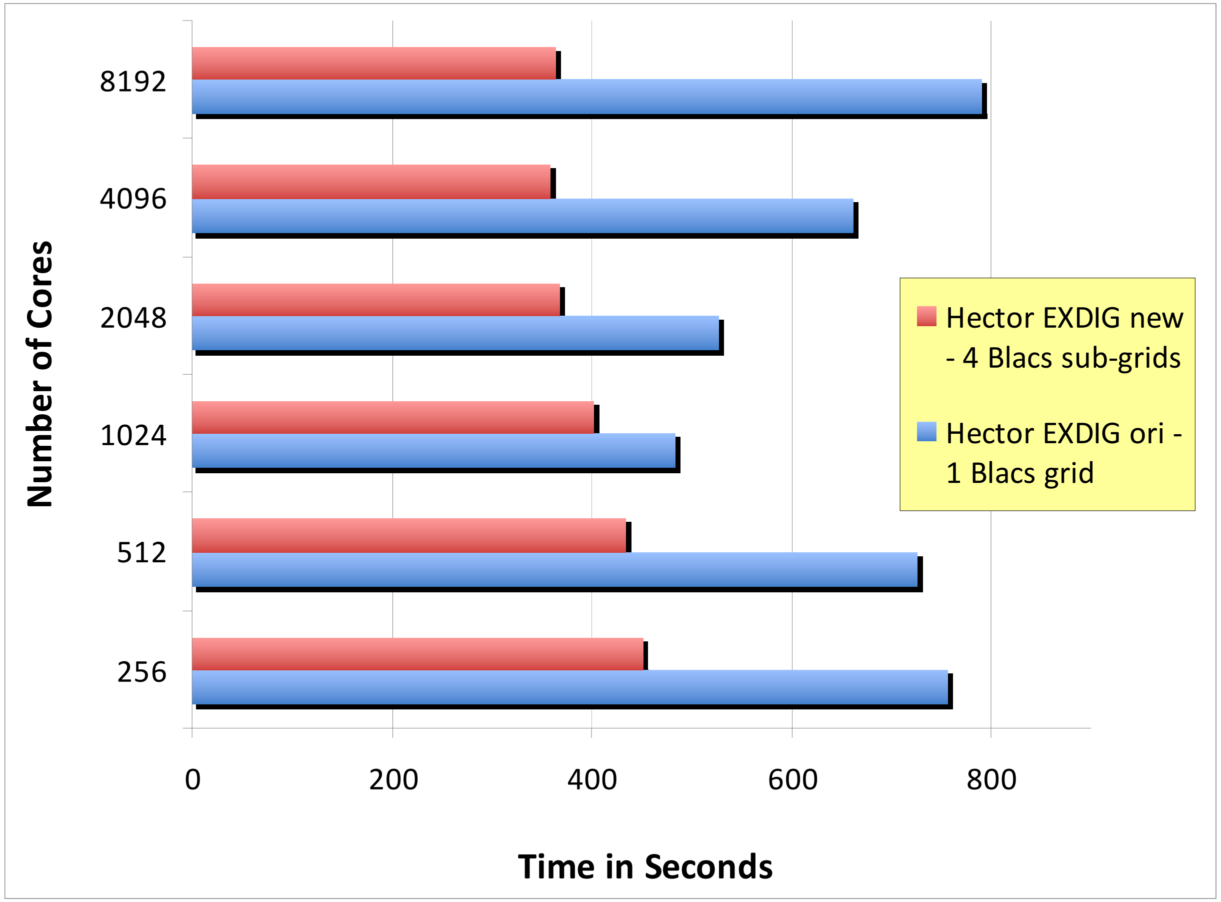
FeIII cases



Parallel Scaling is relatively poor on high core counts unless the problem size is large. Splitting the global Blacs-grid into sub-groups for the sector diags helps mitigate this performance tail-off.

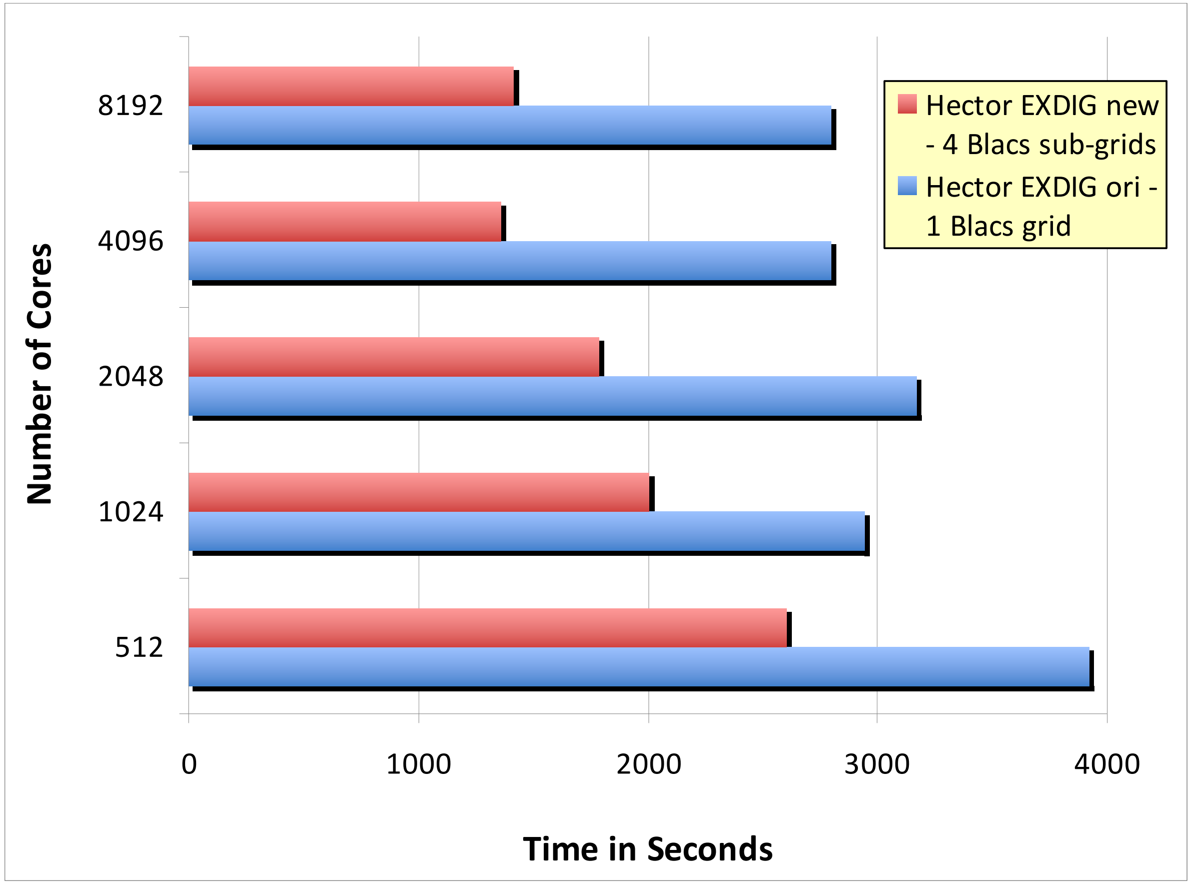
Optimized EXDIG performance

FeIII case, NHAM = 11810



Optimized EXDIG performance

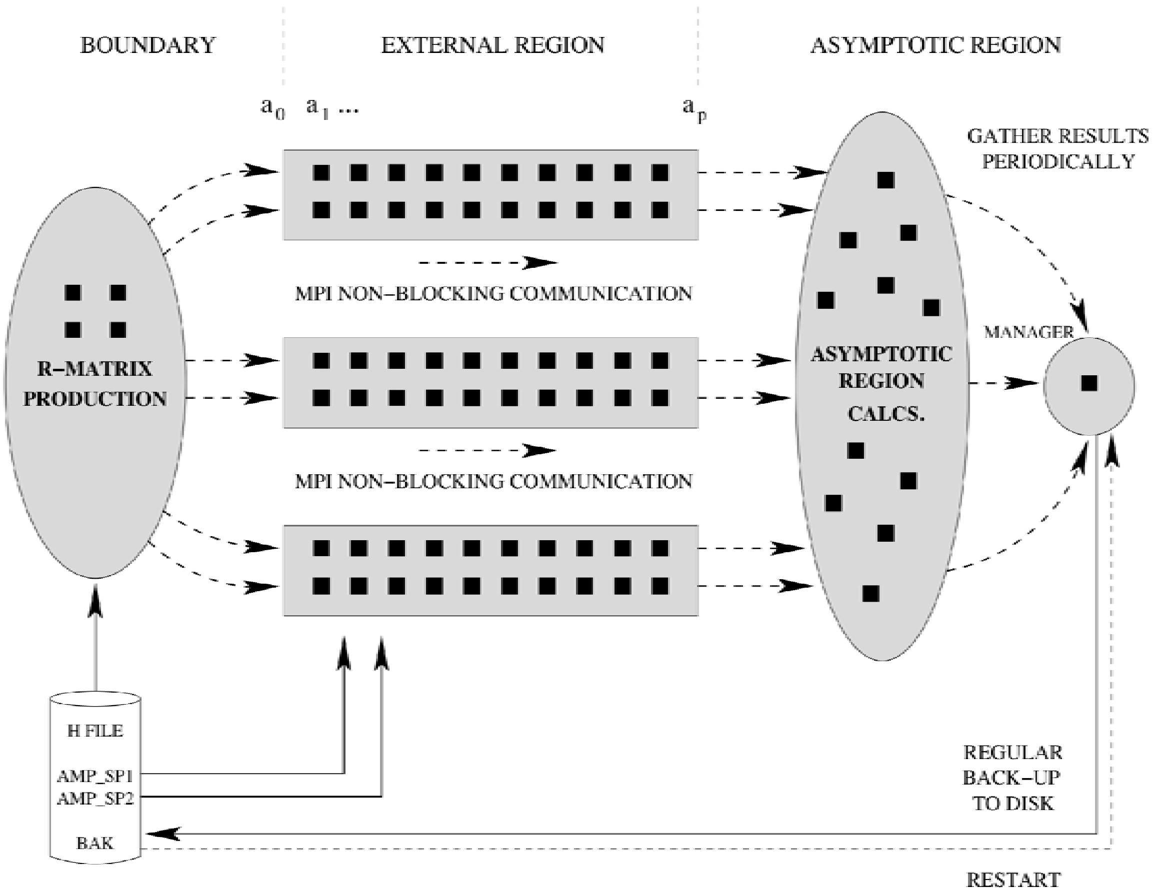
FeIII case NHAM = 44878



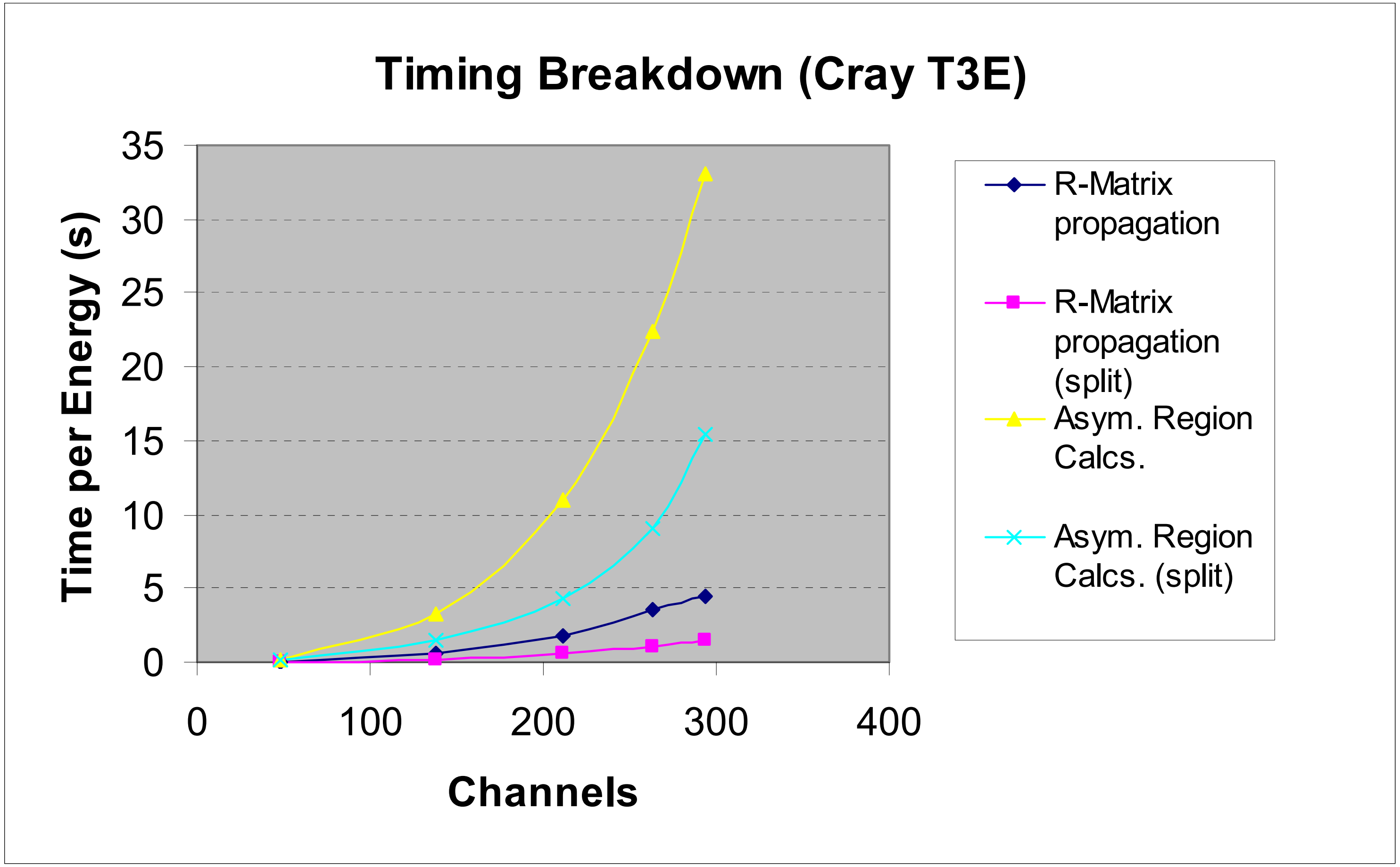
Optimization (ii): Load-balancing PFARM for the XT4

- Load-balancing the functional groups is key to good parallel performance in the propagation stage (EXAS).
  - **Initial R-matrices** need to be **produced** at a sufficient rate to satisfy **demand** from the processor pipelines.
  - Asymptotic calculations must be processed at a sufficient rate to deal with the **supply** of **Final R-matrices** from the processor pipelines.
  - Pipelines must not be held up!
- The size of each group is currently determined by the user (difficult to judge).
- Basic wrapper Perl scripts have been developed to help automate this process, based on performance analyses of the functional groups with different problem sizes.
  - Performance analyses were based on the T3E and need updating for the XT4
  - More sophisticated automation required

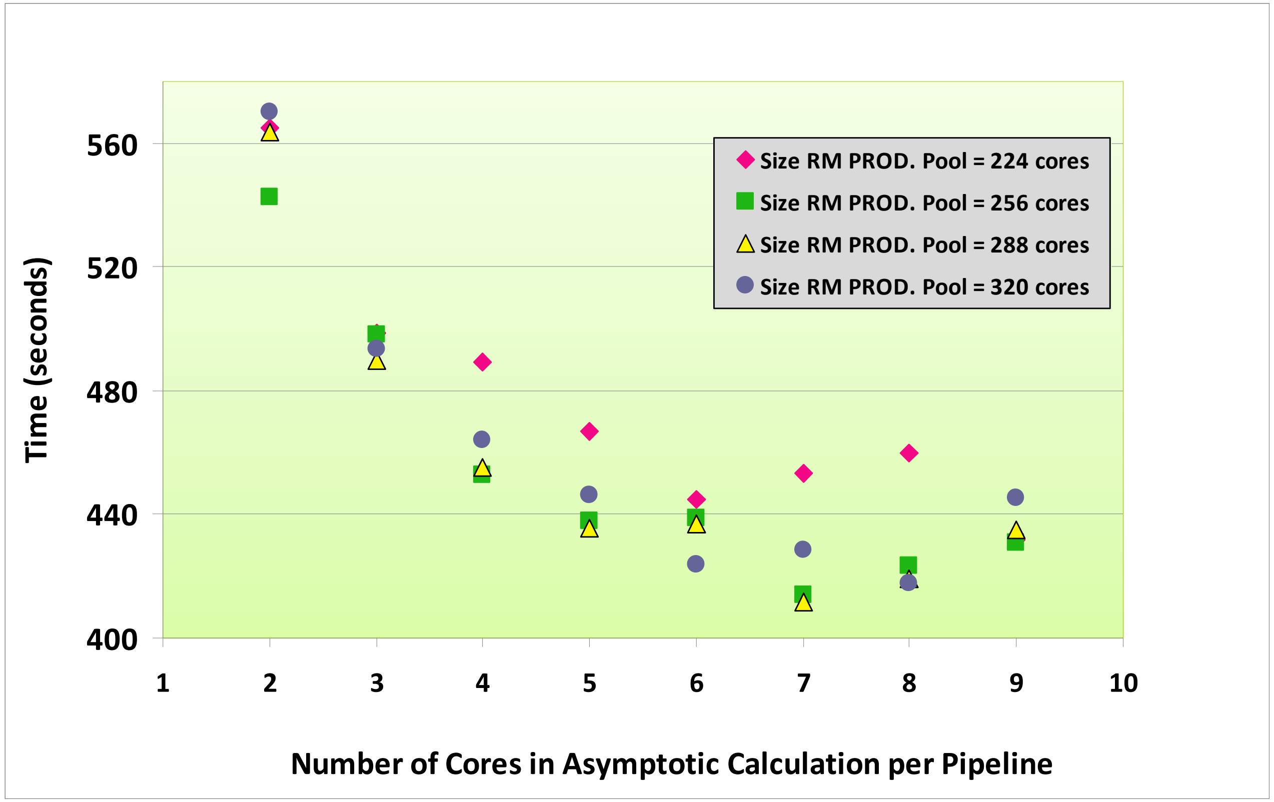
EXAS Parallelization



Example: Load-balancing the pipelines with the asymptotic pool

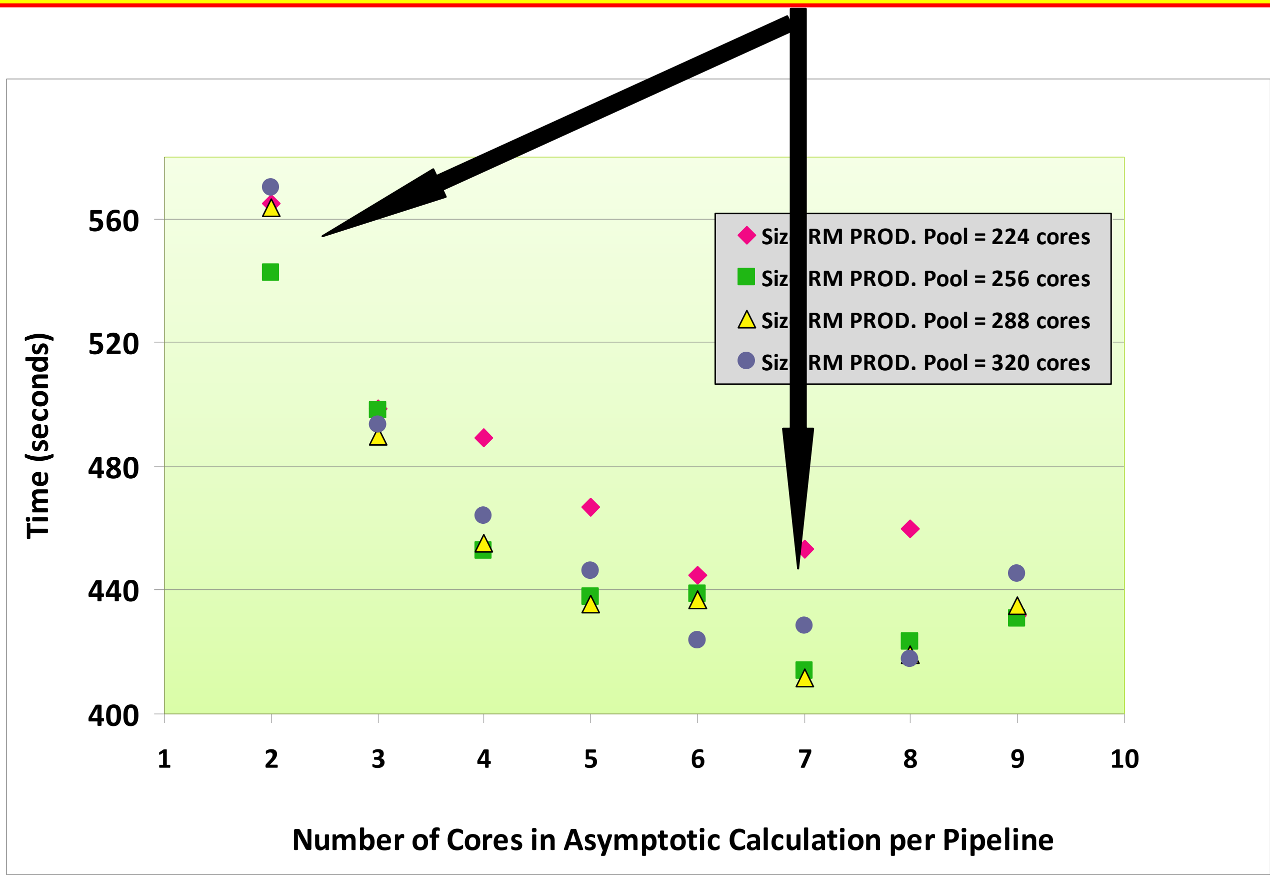


Load-balancing a 1024 core job on the XT4



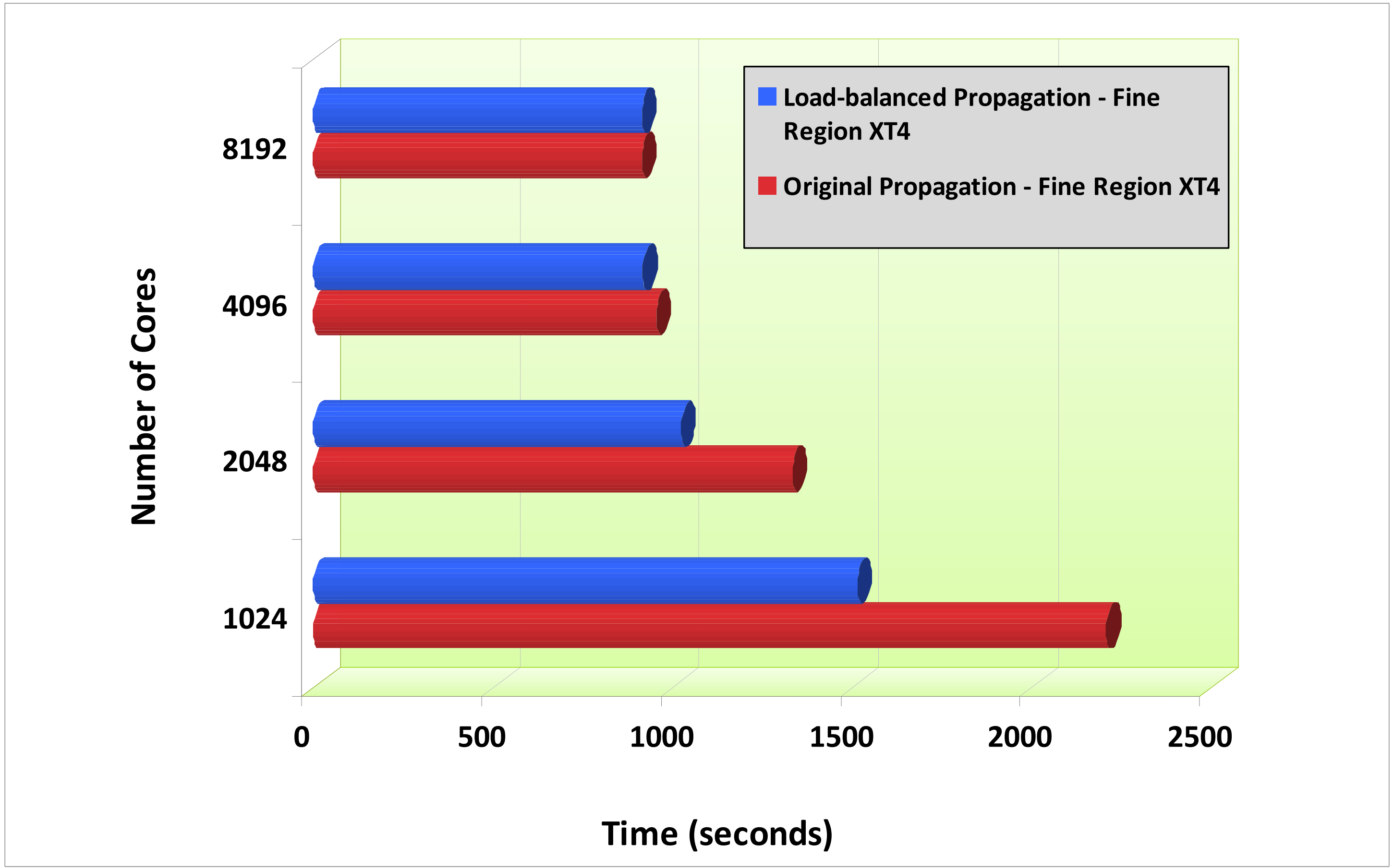
Load-balancing a 1024 core job on the XT4

Upto 30% improvement in speed from correct load-balancing



Load-balanced EXAS performance on the XT4

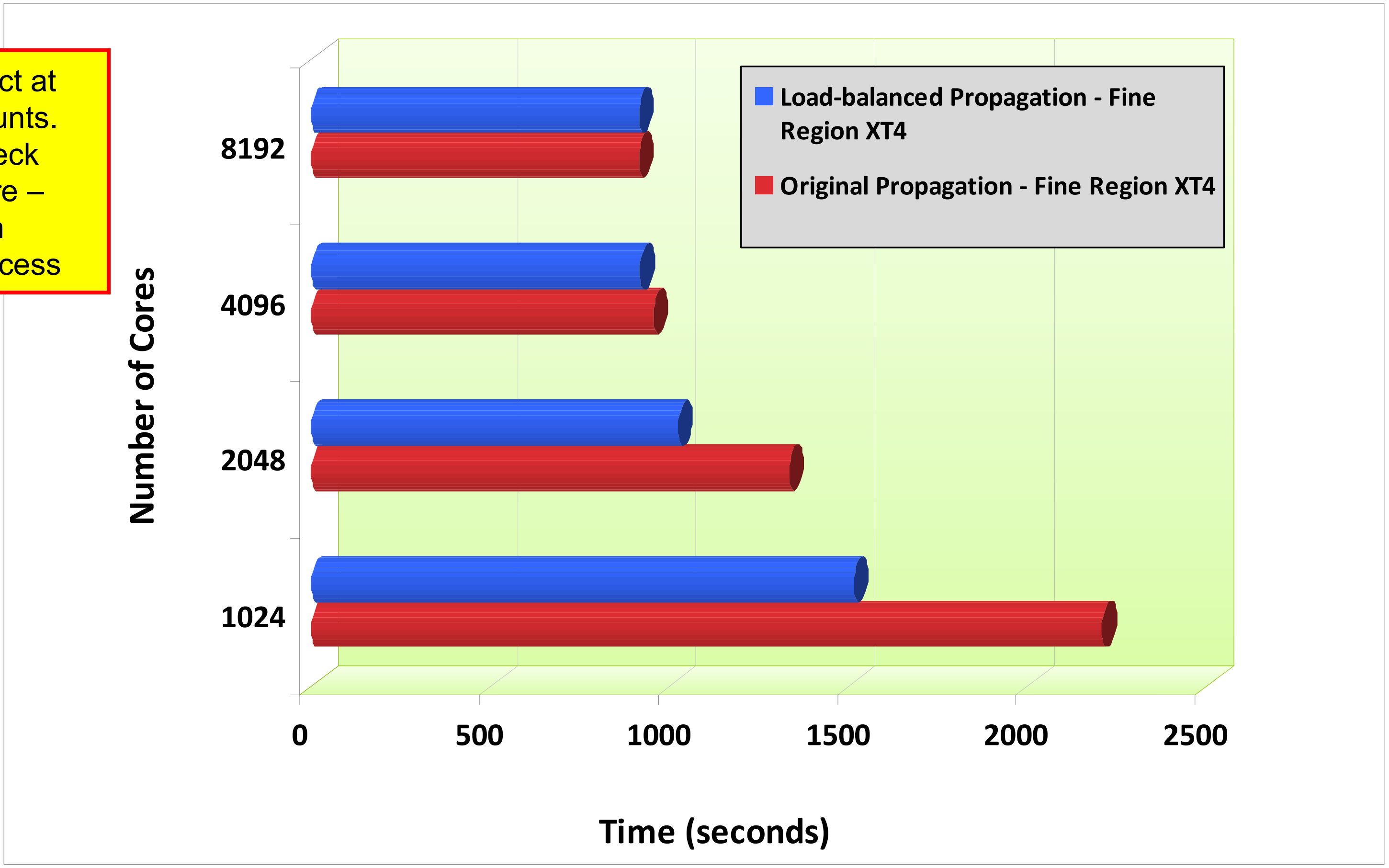
Fine region propagation for FeIIIJ030, 10667 scattering energies



Load-balanced EXAS performance on the XT4

Fine region propagation for FeIIIJ030, 10667 scattering energies

Limited impact at high core counts. Here bottleneck lies elsewhere – probably with manager process



Future Calculations with EXDIG

- More ambitious calculations (more channels, longer range and more complex potentials (particularly for molecules and applications to multiple scattering) imply :
  - More sectors and larger sectors
  - Larger BBM basis sets (and sector Hamiltonians)
- The EXDIG optimizations are designed to ensure that these more ambitious calculations will improve scaling.
- Automate choice of number of sectors in each farm (hence the number of PEs for each ScaLAPACK diagonalization) to produce optimal combination of ScaLAPACK performance and concurrent multiple diagonalizations

Future Work – Propagator Code (EXAS)

- Further automate load-balancing on XT4
  - Based on runs involving a wide range of problem sizes
- Pipelines set-up currently restricted to one sector calculation on one process (legacy of limited memory availability on past machines)
  - Redesign to allow > 1 pipeline sector per core to reduce communication
  - Map efficiently to multi-core architecture to reduce communication
- I/O is now a substantial bottleneck
  - Introduce MPI/IO where appropriate
  - Cliff Noble has developed an XStream library based on MPI/IO with XDR binary format
  - Introduce new sub-manager communicator (i.e. more than one manager process) for gathering and writing final results
- Interface Airy LD propagator method with existing code

Acknowledgements

Christian Voemel, ETH Zurich

Val Burke & Phil Burke, Queen’s University Belfast

Christopher Mueller, STFC Daresbury Laboratory